POZNAN UNIVERSITY OF TECHNOLOGY

Wojciech Mioduszewski

Klasyfikacja danych opisanych za pomocą szeregów czasowych

Master's Thesis

Supervisor: dr inż. Jerzy Błaszczyński

Poznań, 2015

Contents

1 Wstęp **3**

2 Background **5**

2.1 Section . . . 5
2.1.1 Subsection . . . 5
2.2 Inline formatting . . . 5
2.3 Special characters . . . 5
2.4 Figures . . . 6
2.5 Tables . . . 6
2.6 Source code examples . . . 7
2.7 Math . . . 8
2.8 Algorithms . . . 9
2.9 Bibliography . . . 9

3 Concept and Design of the System **11**

4 Implementation **13**

5 Performance Evaluation **15**

6 Conclusions **17**

A Users Guide **19**

Bibliography **21**

Chapter 1

Wstęp

The Introduction may be put *before* the `\mainmatter` command which will disable numbering of this chapter while still adding to the table of contents.

The goal and the scope of the thesis

Chapter 2

Background

The thesis can be structured using the following sectioning styles:

2.1 Section

2.1.1 Subsection

2.1.1.1 Subsubsection

Paragraph

Subparagraph

2.2 Inline formatting

We suggest using *Insets*, like:

strong for strong emphasizing some text.

emph for emphasizing some text.

`Code` for formatting of names of modules, procedures, class names, variables, etc.

`path` for formatting of file names and directories, like `/usr/share/doc/packages/texlive-latex`. The names are properly broken at the ends of lines. However, path names containing special LaTeX characters must be typeset using ERT and the `\dcspath` command, e.g. `sample_file`.

kbd for formatting of shortcuts, e.g.: **Ctrl-c**.

`cmd` for formatting system commands.

`name` for formatting other special names.

2.3 Special characters

1. Non-breaking space can be inserted using **Ctrl-space**. It produces "~" in LaTeX code.

2. A normal, inter-word space can be inserted using **Ctrl-Alt-space**. It produces "`\ `" in LaTeX code. This type of space is useful for formatting spacing after dots, e.g. here. By default LaTeX produces here a longer space used for separating whole sentences.
3. A thin space can be produced by **Ctrl-Shift-space**, e.g.here. It produces "`\,`" in LaTeX code.
4. Sentence-ending space can be inserted using **Ctrl-.**, which produces "`\@.`" in LaTeX code. This type of space is useful in sentences ending with a capital letter. In such cases LaTeX recognizes the last word as a acronym and places a regular inter-word space instead of inter-sentence space. Consider the following example:

   This can be achieved by using HTTP. This protocol...

5. Hyphenation indicator can be inserted using **Ctrl- –**, which is used for marking possible places of hyphenation, e.g. democracy.

2.4 Figures

The figures should be put in floats, like Fig. 2.1. You can also reference figures using prettyref package like this: Fig. 2.1.

Figure 2.1: Example figure

It is possible to combine several pictures inside one float. Just insert a float inside a float. See Fig. 2.2 for example. Please note the horizontal spacing between subfigures.

2.5 Tables

Tables should have captions above like Table 2.1. Use small sans-serif fonts inside tables.

(a) The first subfigure

(b) The second subfigure

Figure 2.2: Example figure

Table 2.1: Example table

Column 1	Column 2	Column 3
One	1	4
Two	2	5
Three	3	6

2.6 Source code examples

There are a few different methods of including sample codes:

1. Using standard LyX-Code style:

```
#include <stdio.h>

int main() {
  printf("Hello world!\n");
  return 0;
}
```

   Note 1: Empty lines must contain at least one single space to remain visible.

   Note 2: There is no way to activate automatic syntax highlighting inside LyX-Code. However, you can use normal inline formatting inside.

   Note 3: Lyx-Code can contain special characters, so it can be used to produce some ASCII art, e.g.:

```
Test 1 ──╫ Test 2
```

2. By inserting *Program Listing*:

```
#include <stdio.h>

int main() {
  printf("Hello world!\n");
```

```
    return 0;
}
```

Note: By default the `lstlisting` environment does not add any left margin. You can change it by adding `xleftmargin` in the *Settings* ▷ *Advanced* dialog box, e.g.:

```
    procedure sayHello()
```

3. By inserting LaTeX Code (ERT block) and using `codeblock` environment:

```
#include <stdio.h>

int main() {
  printf("Hello world!\n");
  return 0;
}
```

4. The listings package can produce floats by itself. See Listing. 2.1 for example.
5. And finally, You can include code from external file:

```
\documentclass[11pt,a4paper,polish,thesis]{dcsbook}

\usepackage[utf8]{inputenc}
\usepackage{babel}
\setcounter{secnumdepth}{4}
\setcounter{tocdepth}{3}

\begin{document}
```

2.7 Math

Can be put inline like this: $S = \sum_{i=1}^{i=K} x_i^2$ or in dedicated lines:

$$S = \sum_{i=1}^{i=K} x_i^2$$

Listing 2.1: The Hello World program in C

```
#include <stdio.h>

int main() {
  printf("Hello world!\n");
  return 0;
}
```

The equations can be also numbered like equation 2.1.

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{2.1}$$

2.8 Algorithms

Use dcsalg package or directly algorithmicx package.

2.9 Bibliography

The bibliography can be included in the thesis like in this case. You can then cite the publications like this [1]. The other (more professional) solution is to use BibTEX. See *LYX User's Guide* for details.

Chapter 3

Concept and Design of the System

Chapter 4

Implementation

Chapter 5

Performance Evaluation

Chapter 6

Conclusions

Appendix A

Users Guide

Bibliography

[1] A. Tanenbaum. *Operating Systems Design and Implementation*. Prentice Hall, 2006.